

## **AMENDMENTS TO THE SPECIFICATION**

Please replace paragraph [020] with the following amended paragraph:

[020] Figure 1 illustrates the use of a Log 150 for storing configuration change information in accordance with example embodiments of the present invention. Typical of most software applications, Application 105 can maintain configuration information in a Configuration Store 145, which can be retrieved using path 140 when Application 105 is loaded. As changes in the configuration settings for Application 105 are made these changes are recorded and stored in Configuration Store 145 using path ~~[[125]]~~ 135. As one of ordinary skill in the art would recognize, these changes can be made by the user (not shown) or by the Application 105 to enhance performance and user experience. Further, these changes can be made while Application 105 is executed, or the changes can be made directly to the Configuration Store 145, which could be, e.g., an application file or centralized registry. In accordance with the present invention, however, changes are typically made while Application 105 is executed. Nevertheless, changes to Configuration Store 145 can be made when Application 105 is not being executed. For example, when the configuration settings are shared among several applications (*e.g.*, printer settings), changes to one application may effect changes to all other shared applications or changes may be made directly to the Configuration Store 145 through a Configuration File or Registry Editor (not shown).

Please replace paragraph [027] with the following amended paragraph:

[027] In accordance with yet another example embodiment, when any reversion routine is called and then ~~[[passed]]~~ passes Package 110 back to Application 105, the Configuration Store 145 may be updated. As such, Application 105 may ~~[[also]]~~ then send another Package 110 of configuration change information to Log 150. This implementation of the present invention is particularly useful in assisting the redo functionality.

Please replace paragraph [028] with the following amended paragraph:

[028] Because the number of events (*i.e.*, configuration changes) can be large, the present invention allows for distinguishing between events that are interesting to the user or Application 105 and those that are not. For example, the present invention can allow the user to select which events to send to Log 150 through *e.g.*, a dialog box which can be checked if the user wishes to send the change he is making to Log 150. ~~[[In conjunction]]~~ Thus, exemplary embodiments provide that Application 105 can automatically send those events that are visible to the user, *e.g.*, change in font size, printer settings, toolbar configuration, etc. Although Log 150 can receive both internal application settings and visible user event settings, typically Log 150 will be given the ability to distinguish internal events and those that are controlled and visible by the user.

Please replace paragraph [032] with the following amended paragraph:

[032] There are several ways a callback for a particular application can be performed. For example, in order to support reversion, an Application 210 may need to implement a specific interface and routine specified by the interface that ~~[[manage]]~~ all reversions. This routine may

in turn call other routines when reverting to a previous configuration setting using the information in Package 220, but the callback initially will be made to the specific managing routine. If Application 210 does not support this managing routine, an Error Code 215 can be raised when the callback is attempted. Alternatively, the callback routine can be encoded inside Package 220, as, for example a function pointer, callback or delegate for loading the reversion routine, with various data also encoded inside Package 220 passed as arguments.

Please replace paragraph [035] with the following amended paragraph:

[035] As one of ordinary skill in the art would recognize, rather than sending an Error Code 215, Application 210 can also send a success code (not shown) to Change Log 205 upon a successful configuration reversion. Accordingly, Change Log 205 can note this change as part of the history for future revisions such as redoing an undo operation.

Please replace paragraph [036] with the following amended paragraph:

[036] Figure 3 illustrates example steps and acts used in reverting a current configuration setting for an application to a previous configuration setting. A step for Maintaining [[310)] 310 a history of configuration setting changes for an application may include acts for Receiving [[312)] 312 application configuration information and Storing [[314)] 314 this information. This application configuration information is to be used in reverting to the previous configuration setting. Further, this configuration information could be Received [[312)] 312

and Stored ~~[(314)]~~ 314 as a result of a user request or automatically whenever a configuration change is made to the application.

Please replace paragraph [038] with the following amended paragraph:

[038] A step for Identifying ~~[(320)]~~ 320 application configuration information from within the maintained history may include an act of Displaying ~~[(322)]~~ 322 a representation of the application configuration information within a UI for viewing and selecting the application configuration information. Such UI may include, *e.g.*, a filtering device that would allow for filtering results, *e.g.*, on a per application or per time period basis. The information displayed might then be a Name of the application whose configuration information changed, a change Title, Description, or Date and Time of the change. This information may be displayed in a grid-like fashion with rows or columns or any other suitable format. As previously mentioned, the information could be displayed for an individual application, or chronologically from a history of multiple applications.

Please replace paragraph [039] with the following amended paragraph:

[039] A step for Reverting ~~[(330)]~~ 330 to the application's previous configuration setting using a portion of the identified application configuration information may include acts of Retrieving ~~[(332)]~~ 332 the application configuration information and Calling ~~[(334)]~~ 334 a reversion routine. The configuration information can be passed to the routine for reverting to the application's previous configuration setting. This routine may display a link that gives the user

instructions on actions or steps to perform in order to revert the application to the previous configuration settings. Alternatively, the routine may be used in automatically reverting to the applications previous configuration setting in accordance with those example embodiments described above. The reversion to the application's previous configuration setting could be an undo, redo or roll-back operation as also described above. The routine may be embedded within the header of the configuration information or the header could comprise a pointer to the reversion routine called. Further, the revision routine could call one or more other routines for reverting to the applications previous configurations setting.

Please replace paragraph [040] with the following amended paragraph:

[040] Figure 4 illustrates example acts used in facilitating the reversion of an application's current configuration setting to a previous configuration setting. The system provides for the act of Receiving <sup>410</sup> a package of information, which is to be used in reverting to the previous configuration setting. The package of information can then be Stored <sup>420</sup> for subsequent retrieval and for maintaining a history of configuration setting changes for the application. A request from the application for the package of information is Received <sup>430</sup> when reverting to the applications previous configuration setting. Finally, the package of information is Retrieved <sup>440</sup> and sent to the application for processing when reverting to the previous configuration setting.